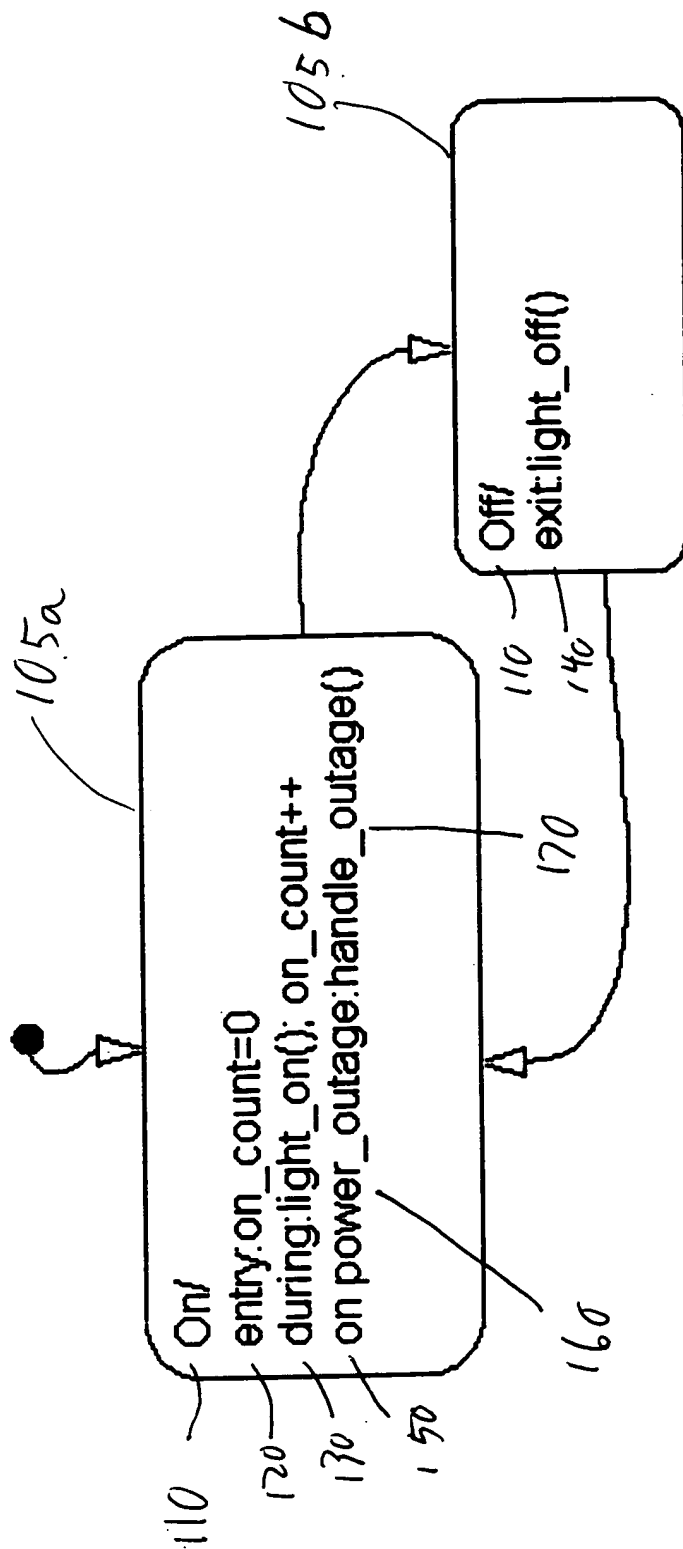


Figure 1

100



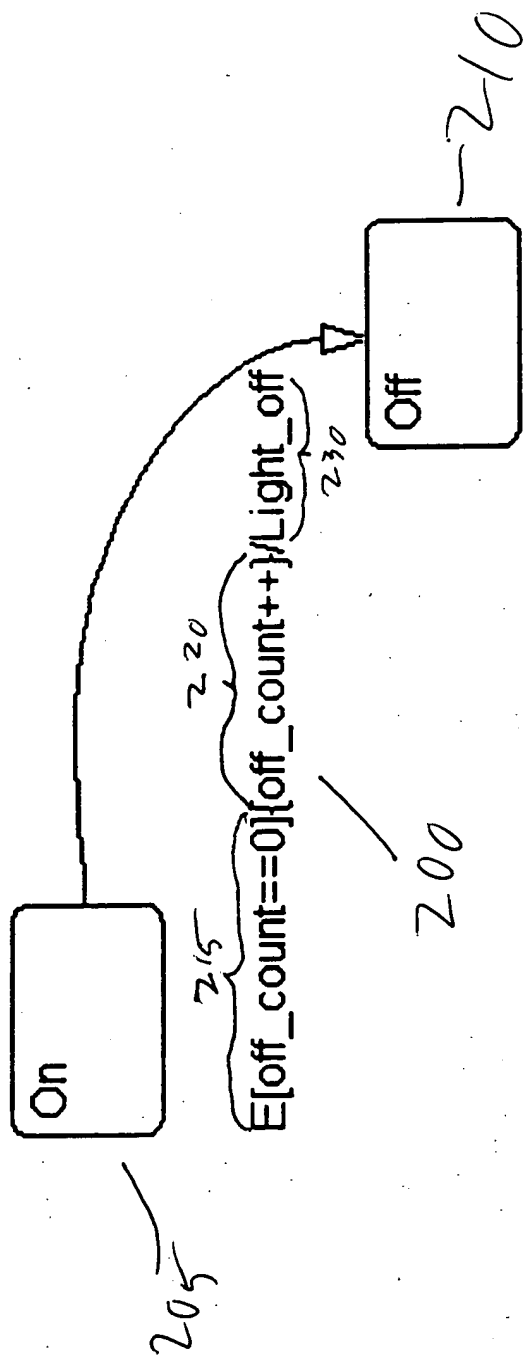


Figure 2

Figure 3

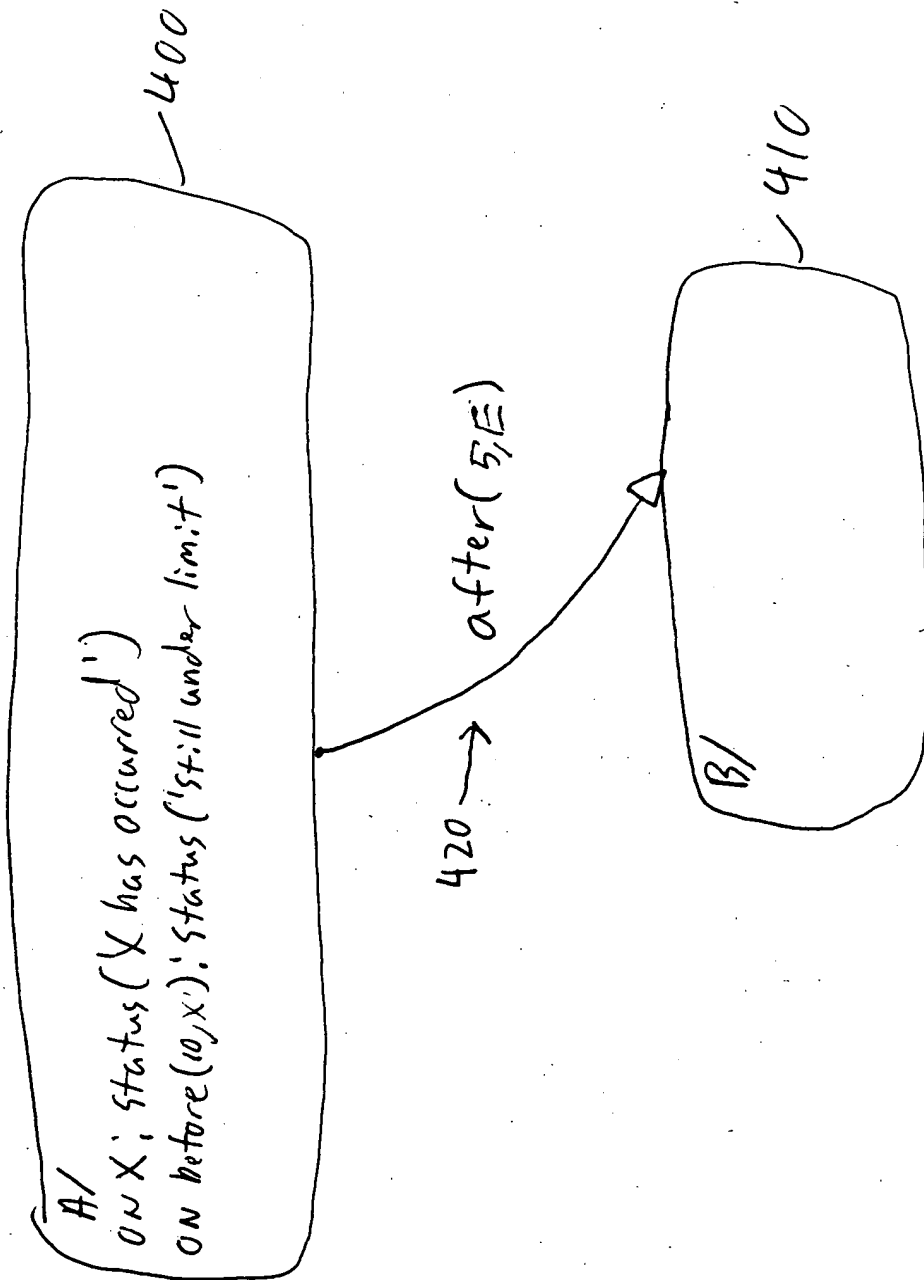
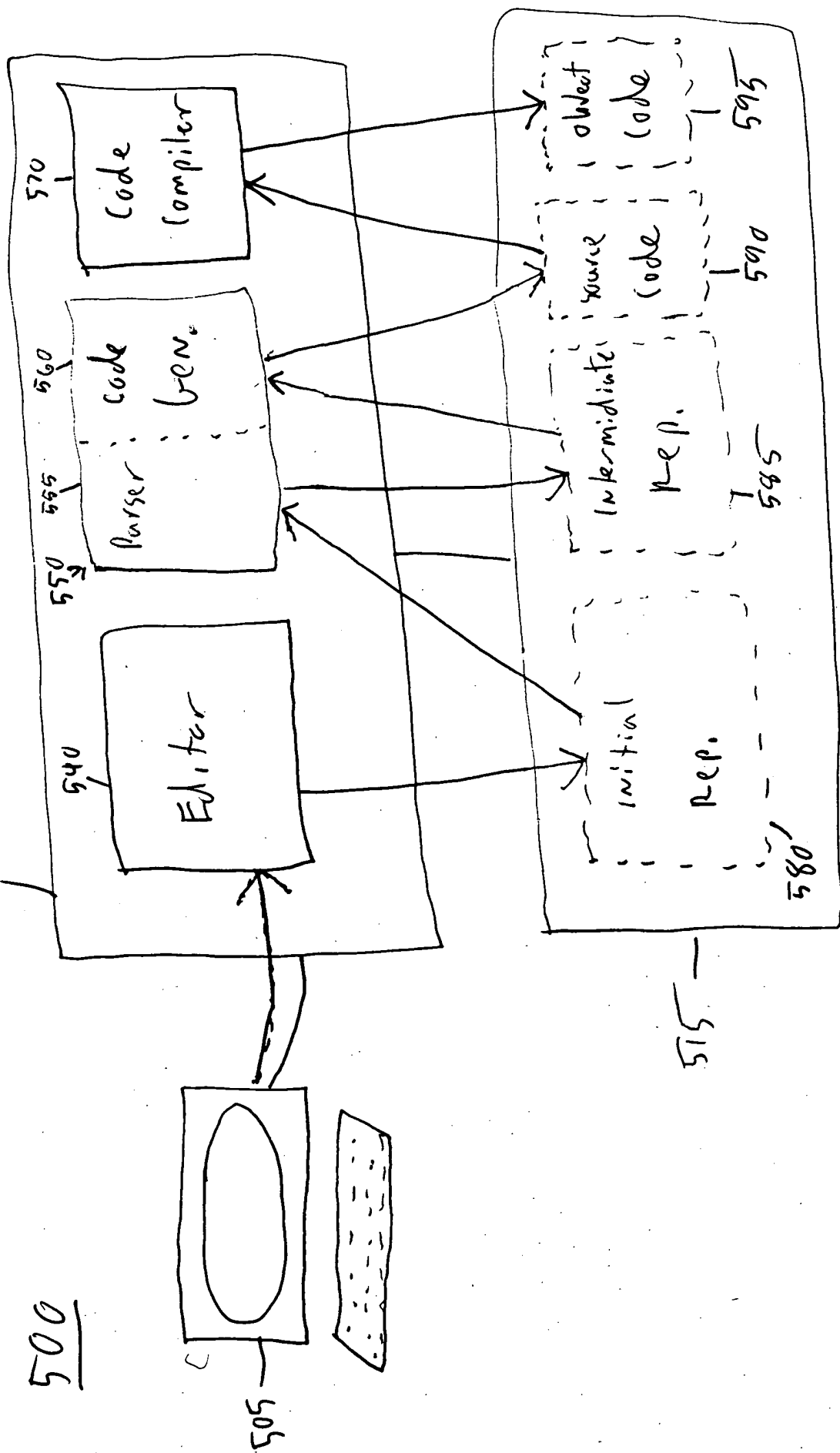


Figure 4

500

510



515

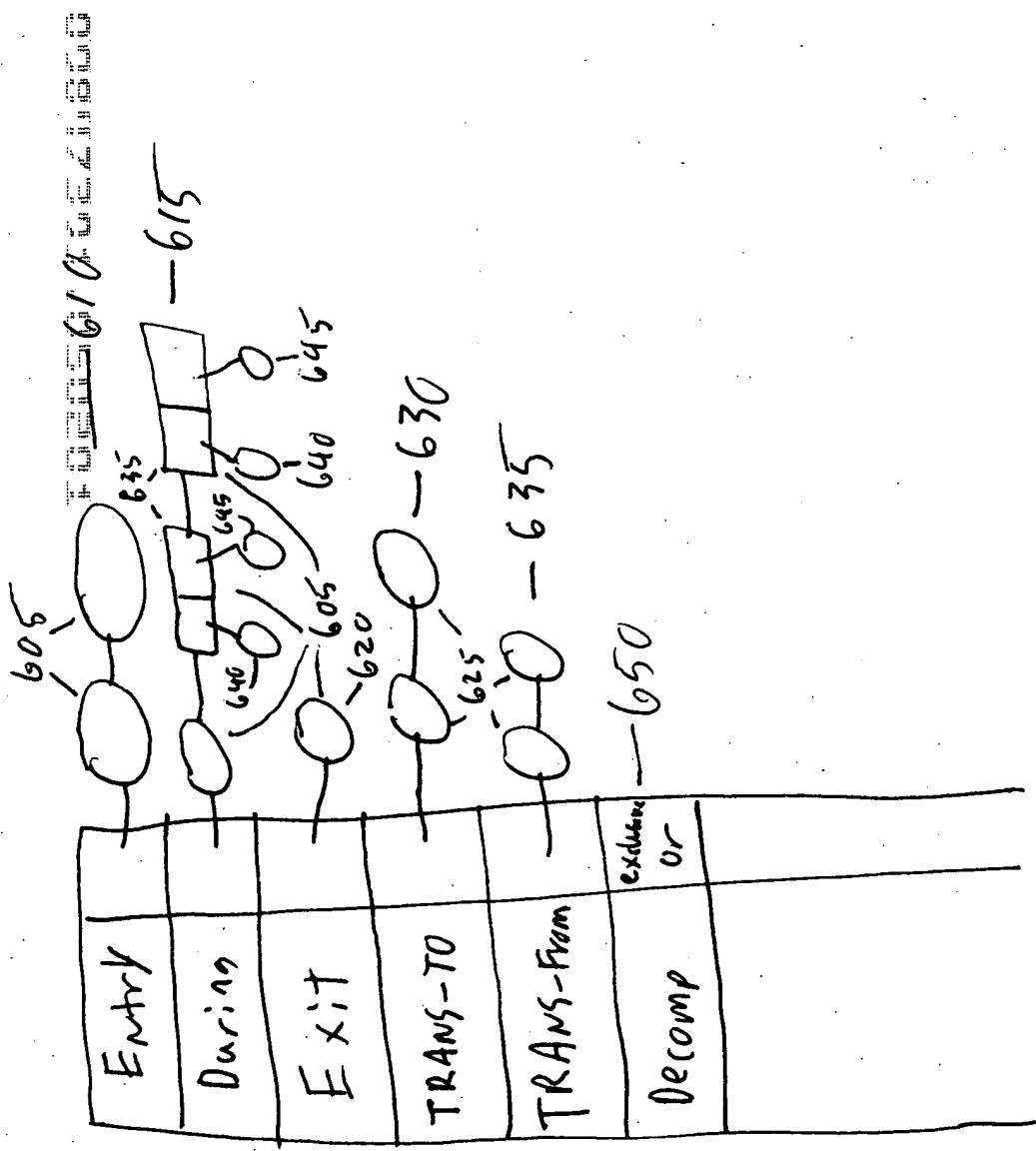


Fig. 5

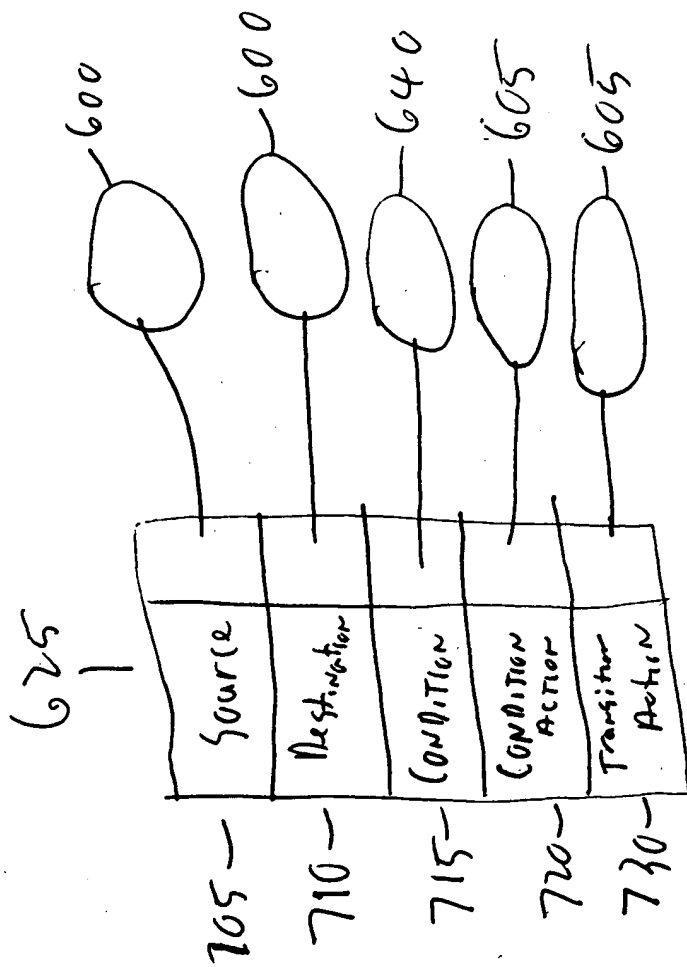


Fig. 6

800

TABLE 1

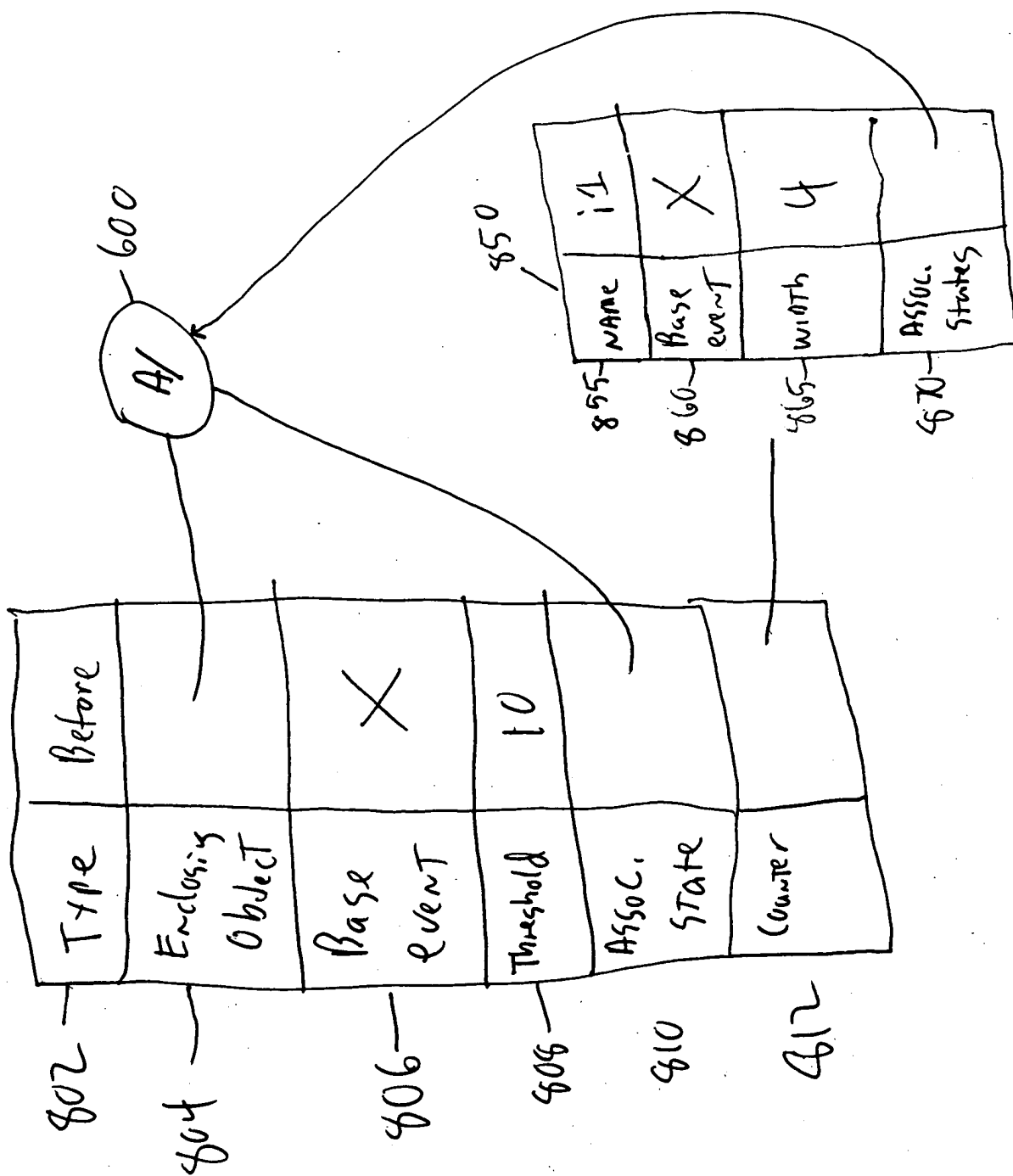


Fig. 7

Fig. 8

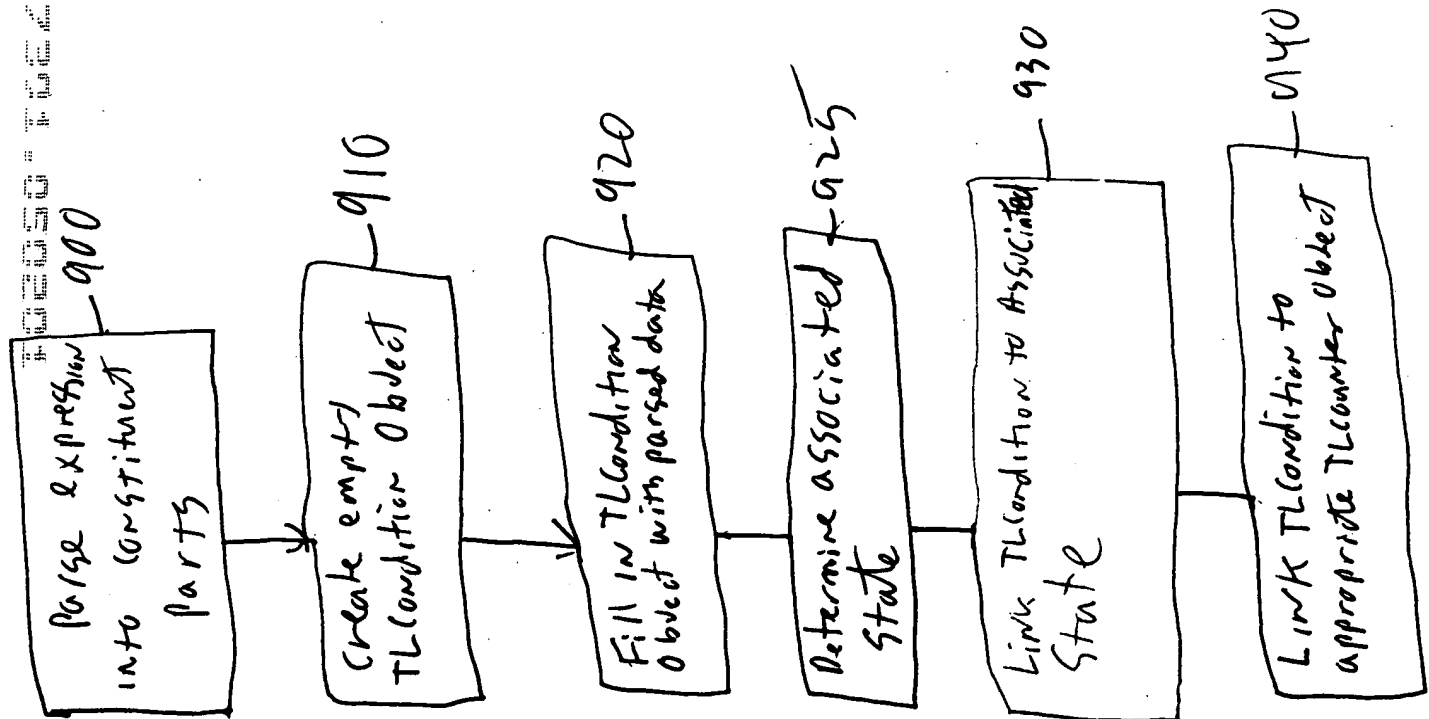


Fig. 9

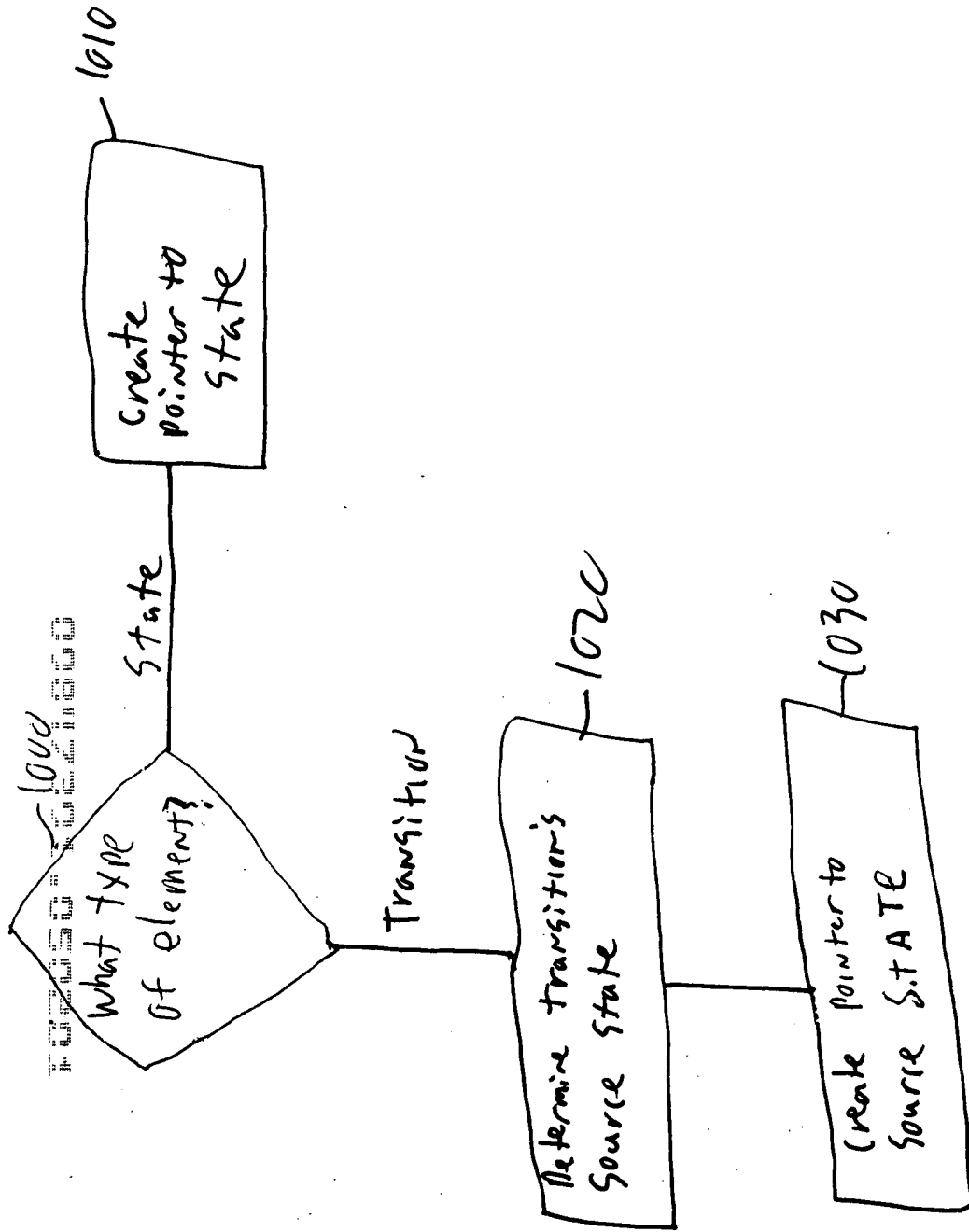


Fig. 10

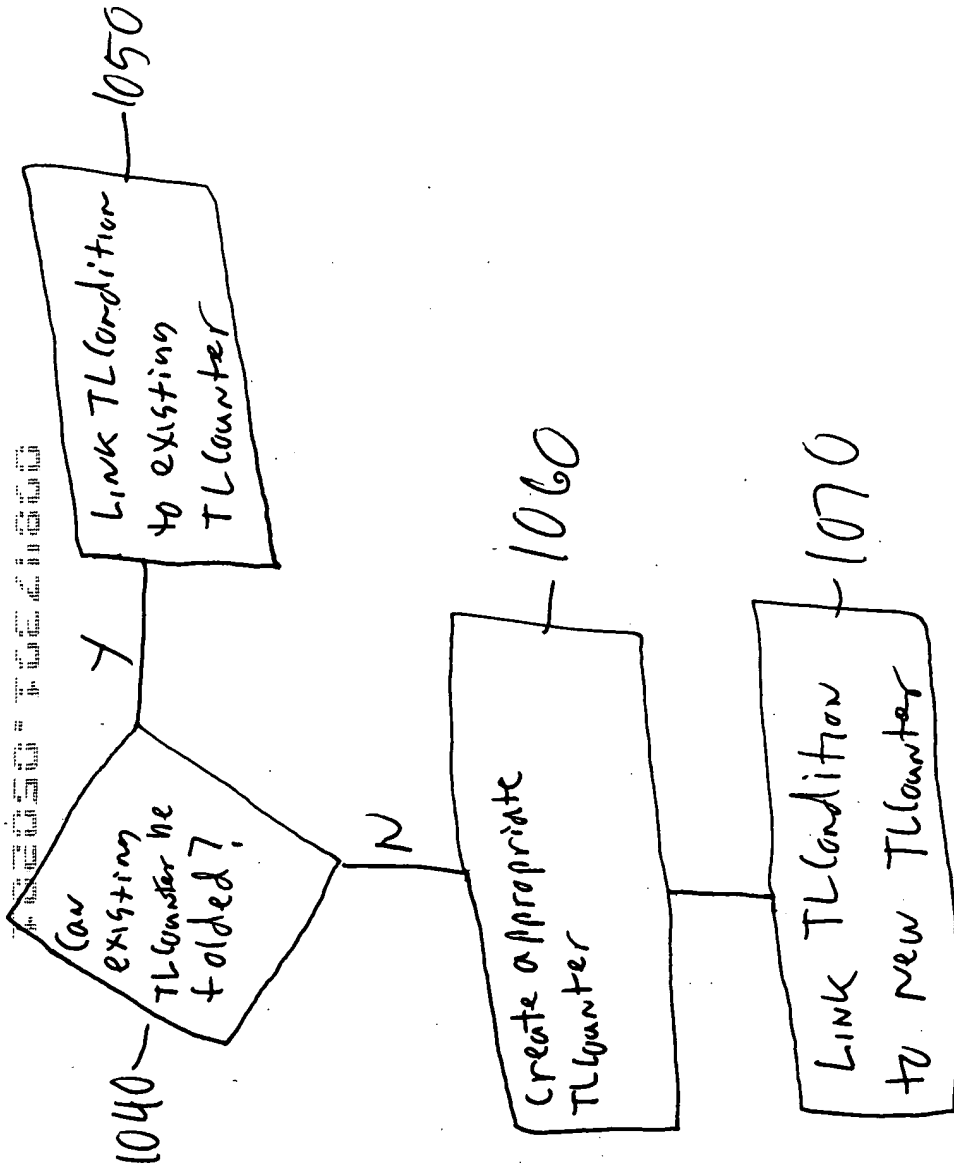


Figure 11

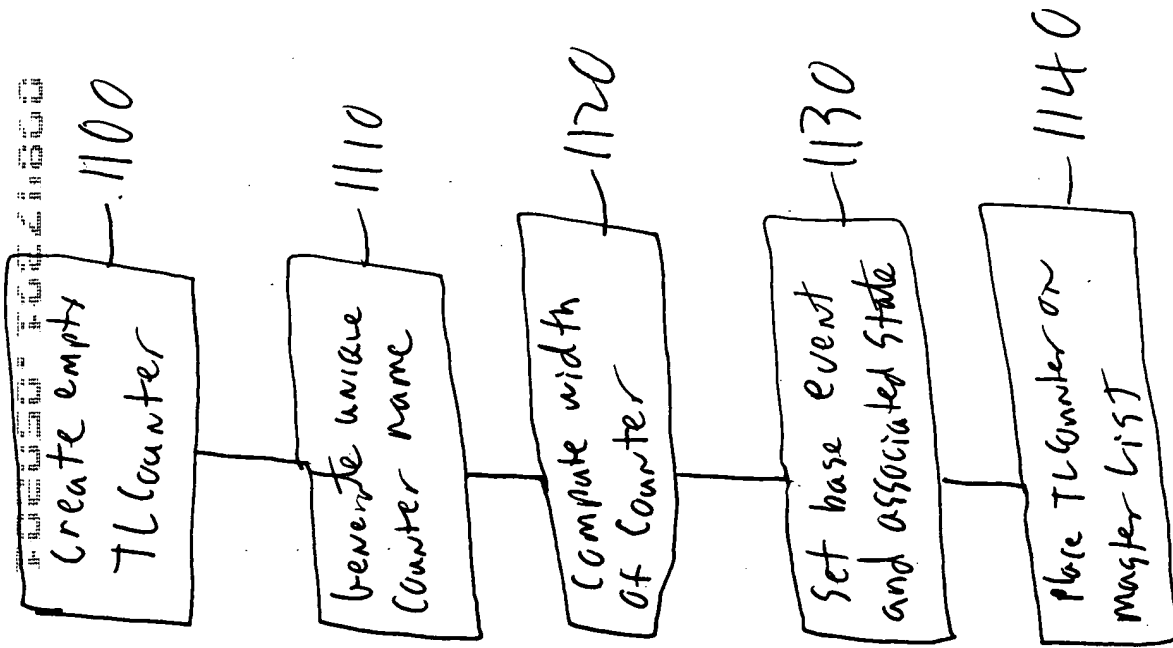


Figure 12

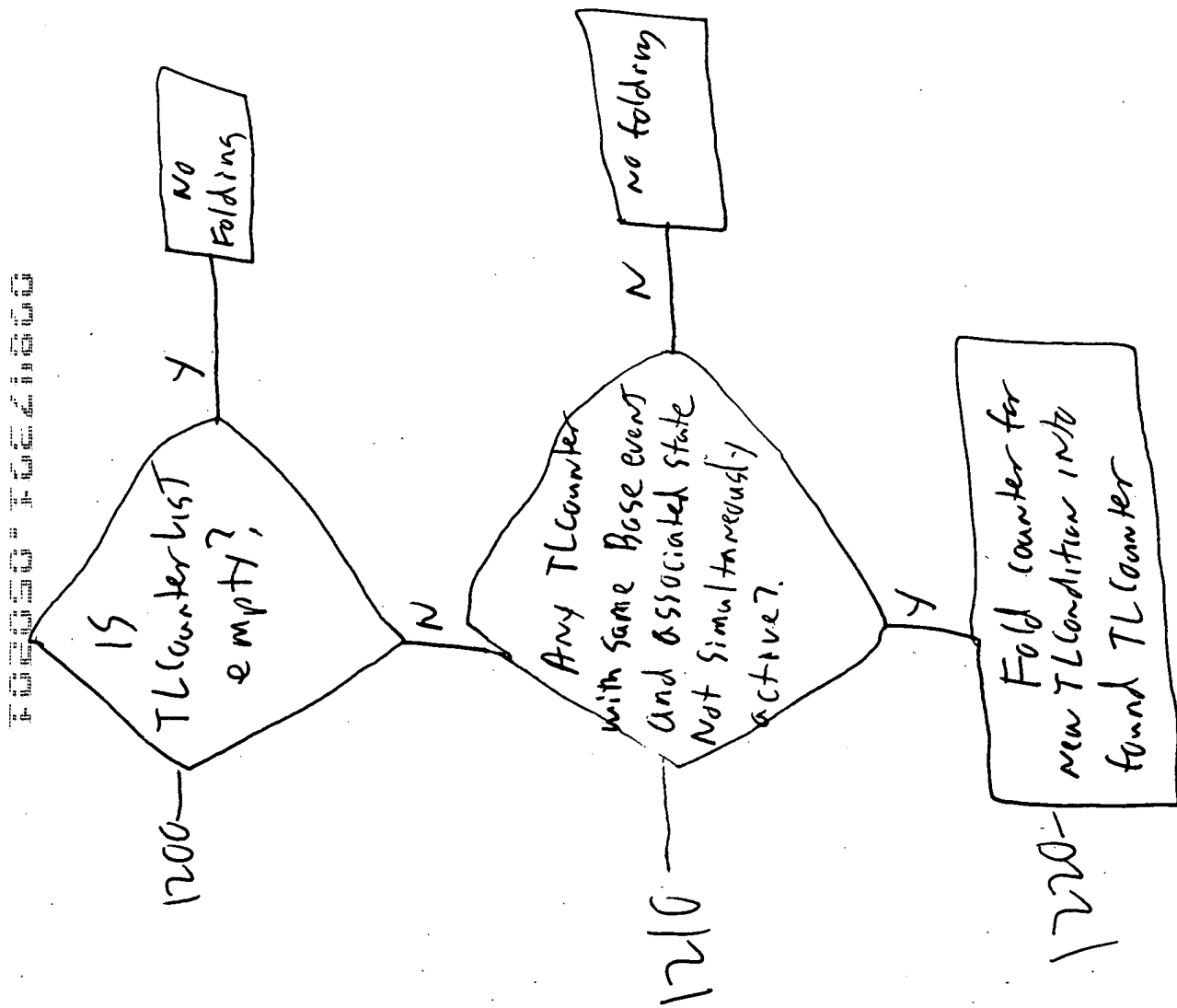


Figure 13

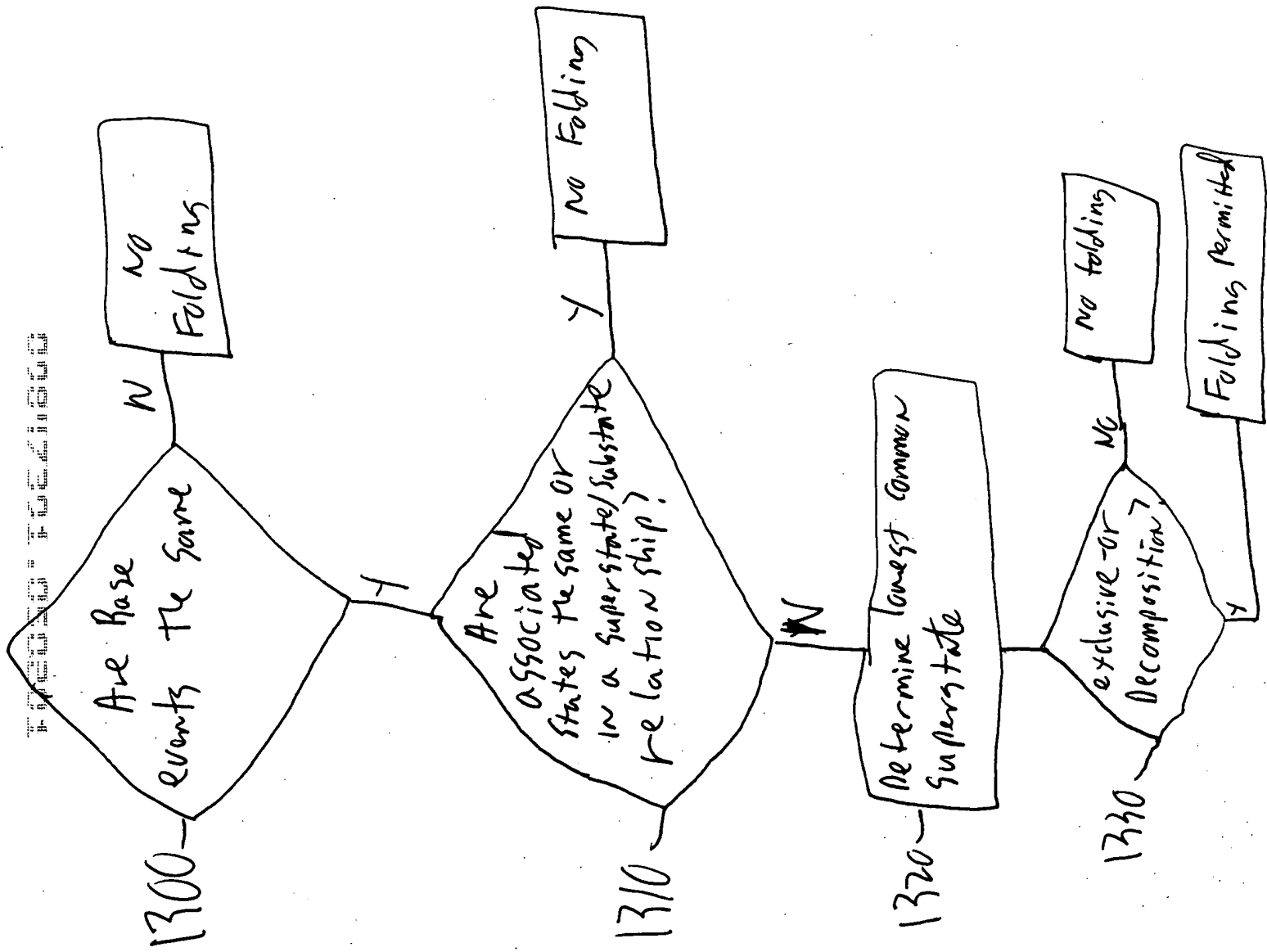


Figure 14

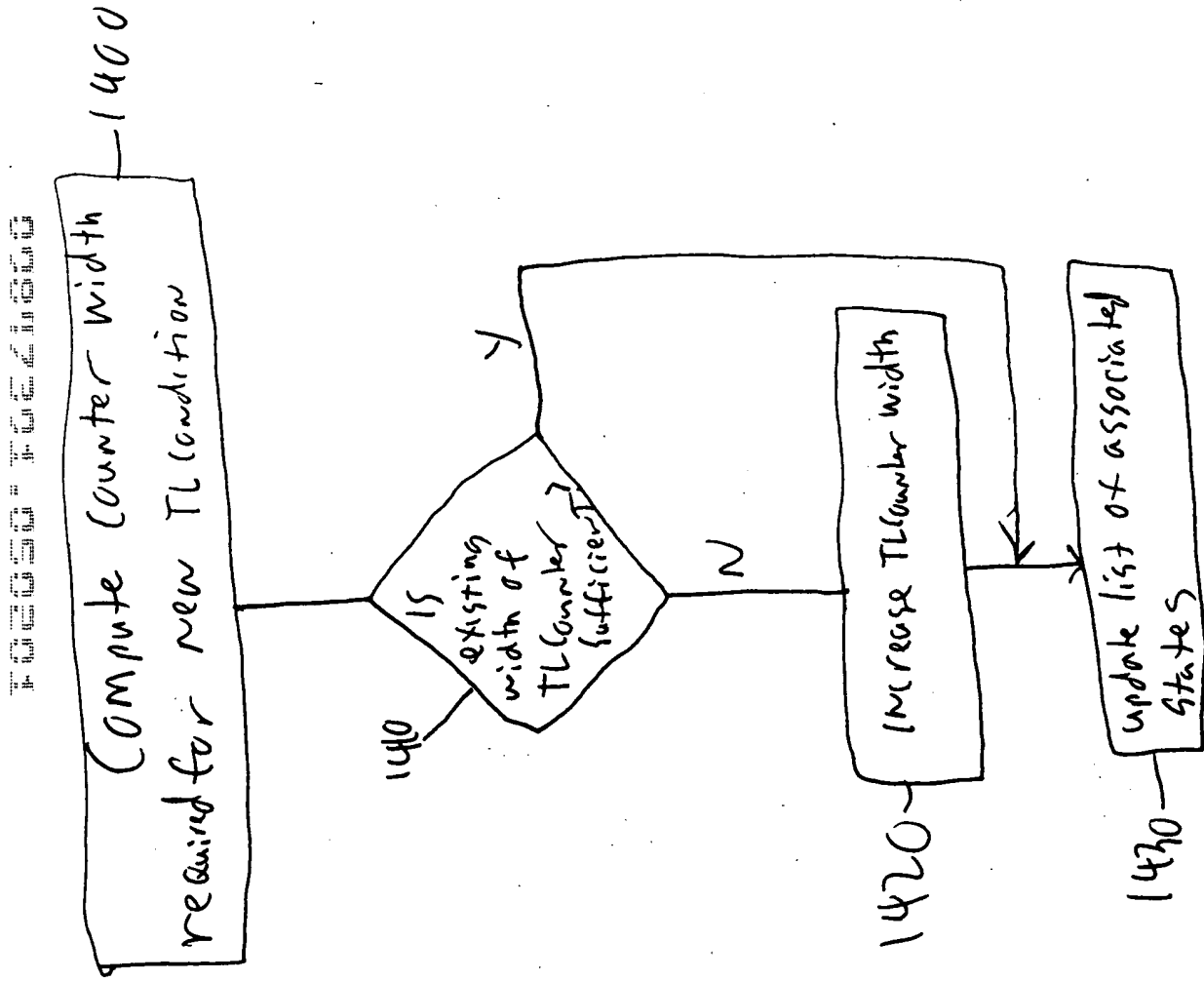


Figure 15

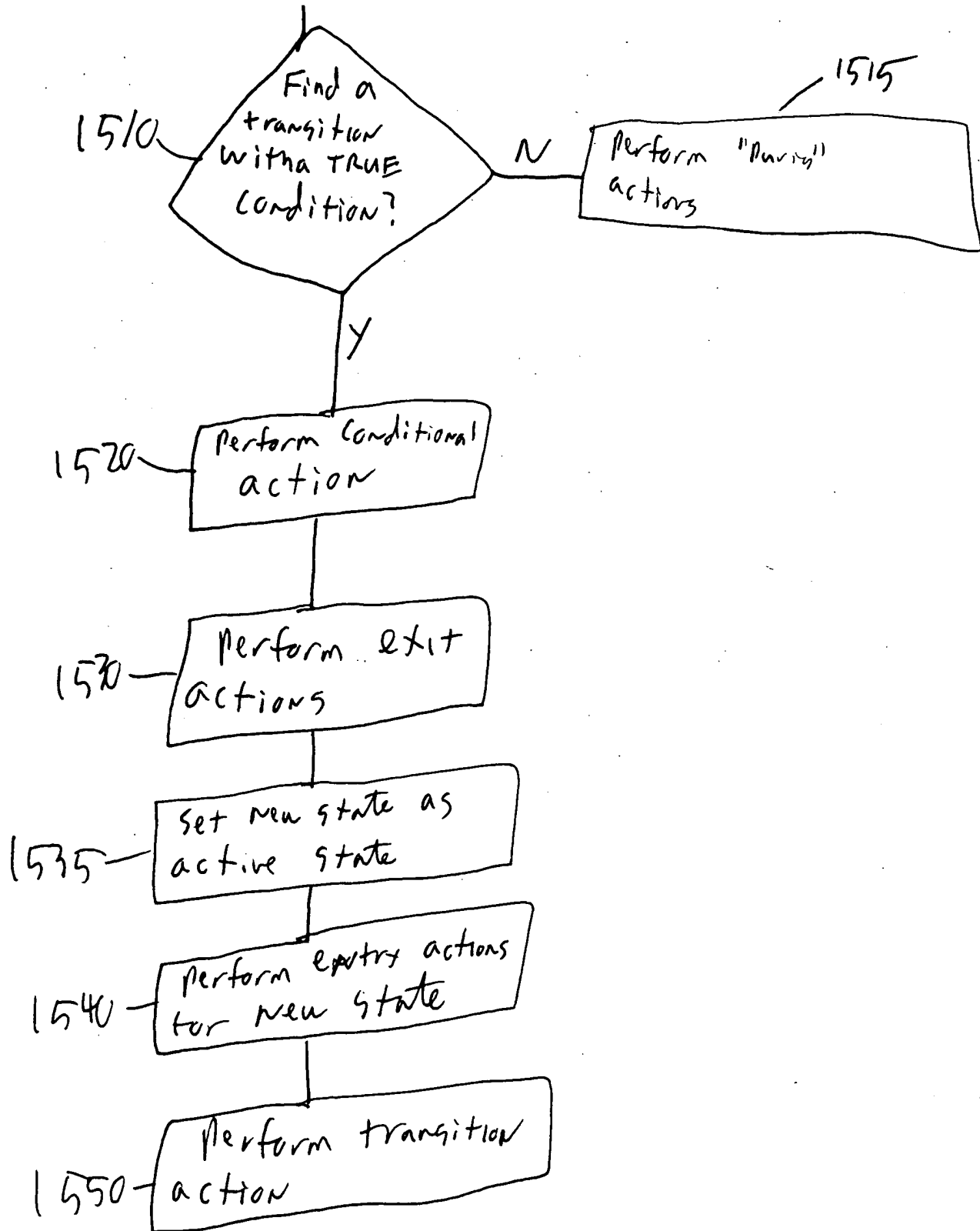
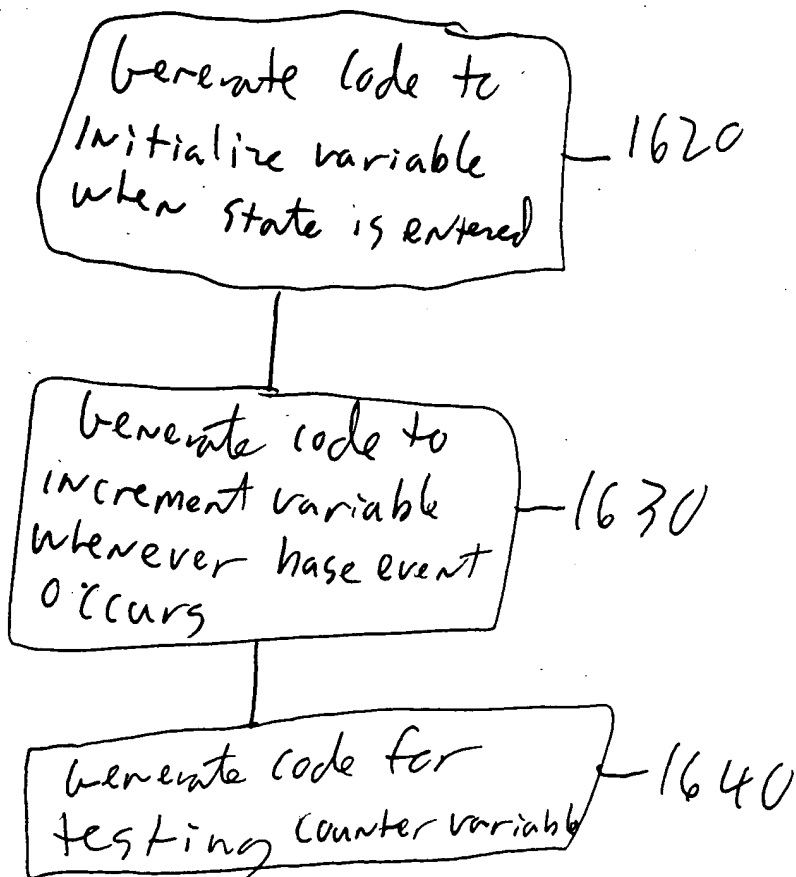


Figure 16



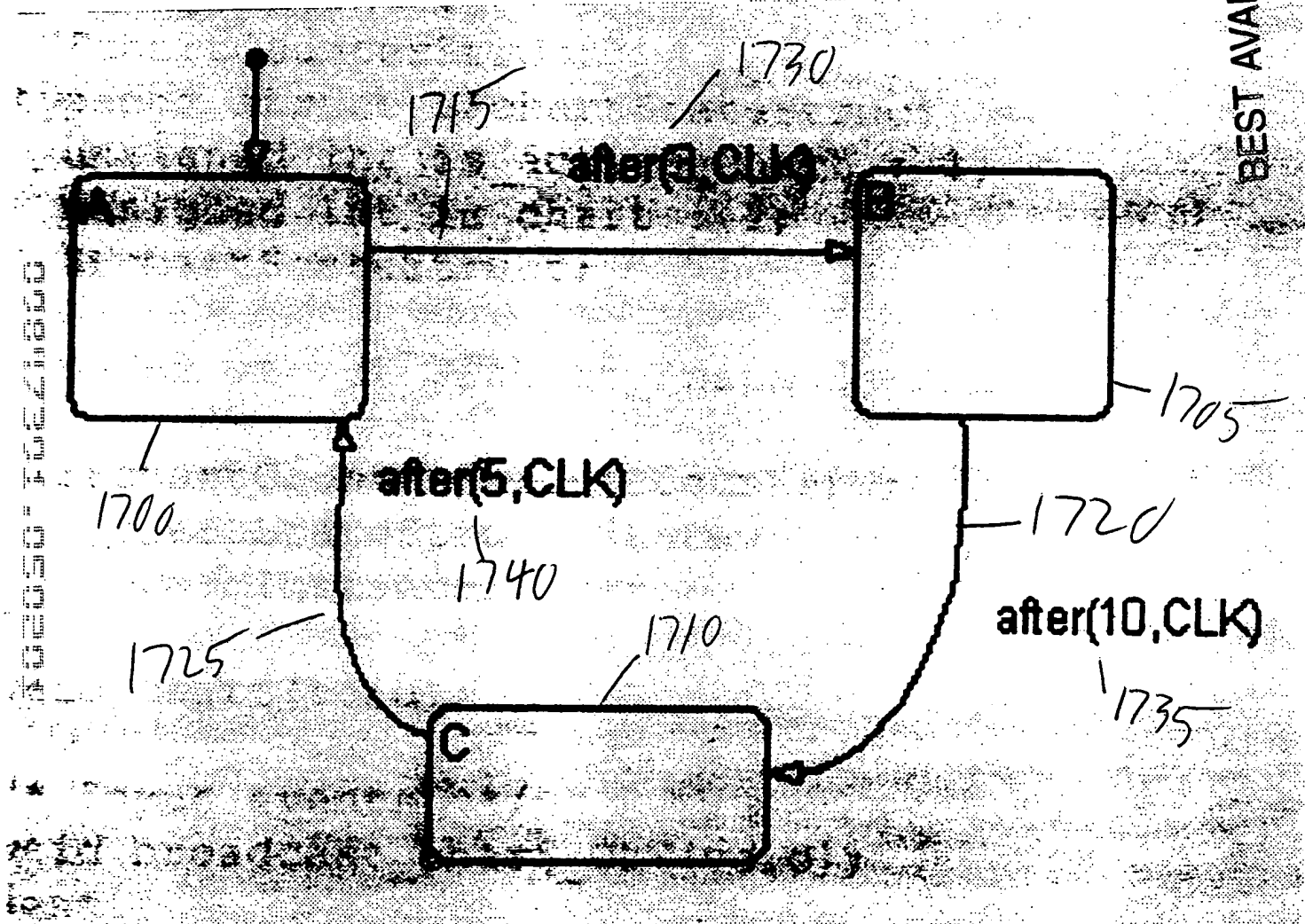


Figure 17

```

/*
 *
 * Stateflow code generation for chart:
 *   temporal_example/Chart
 *
 * Target Name                      : target
 * Model Version                    : 1.188
 * Stateflow Version                : 4.0.3.12.00.1.000000
 * Date of code generation          : 26-Mar-2001 12:31:13
 */

#ifndef __chart_h__
#define __chart_h__
typedef struct SFchartCounterStruct { } 1800
    unsigned int il : 4;
} SFchartCounterStruct;
typedef struct SFchartStateStruct { } 1820
    unsigned int is_active_chart : 1;
    unsigned int is_chart : 2;
} SFchartStateStruct;

typedef struct SFchart_InstanceStruct { } 1830
    SFchartCounterStruct Counters;
    SFchartStateStruct State;
} SFchartInstanceStruct;

void chart(void); } 1840

/* Input events: */
void broadcast_chart_CLK(void); } 1810
#endif

```

Figure 18

```

*      tempo_ example/Chart
*
* Target Name           : target
* Model Version         : 1.188
* Stateflow Version     : 4.0.3.12.00.1.000000
* Date of code generation : 26-Mar-2001 12:31:13
*
*/

```

```

#include "temporal_example_target.h"
#include "chart.h"

```

```

#define IN_NO_ACTIVE_CHILD           (0)
#define IN_cl_sl_A                   1
#define IN_cl_sl_s2_B                2
#define IN_cl_sl_s3_C                3
#define event_CLK                     0
static SFchartInstanceStruct chartInstance;
void chart(void);

```

```

void chart(void)
{

```

```

    /* During: Chart */

```

```

    if(_sfEvent_temporal_example_ == event_CLK) {
        if(chartInstance.Counters.il<0xfU) {
            chartInstance.Counters.il++;
        }
    }

```

```

    if(chartInstance.State.is_active_chart == 0) {

```

```

        /* Entry: Chart */

```

```

        chartInstance.State.is_active_chart = 1;

```

```

        /* Entry: A */

```

```

        chartInstance.State.is_chart = IN_cl_sl_A;

```

```

        chartInstance.Counters.il=0;

```

```

    } else {

```

```

        switch(chartInstance.State.is_chart) {

```

```

            case IN_cl_sl_A:

```

```

                /* During: A */

```

```

                if((_sfEvent_temporal_example_ == event_CLK) &&
                    (chartInstance.Counters.il >= 3)) {

```

```

                    /* Exit: A */

```

```

                    /* Entry: B */

```

```

                    chartInstance.State.is_chart = IN_cl_sl_s2_B;

```

```

                    chartInstance.Counters.il=0;
                }

```

```

                break;

```

```

            case IN_cl_sl_s2_B:

```

```

                /* During: B */

```

```

                if((_sfEvent_temporal_example_ == event_CLK) &&
                    (chartInstance.Counters.il >= 10)) {

```

```

                    /* Exit: B */

```

```

                    /* Entry: C */

```

```

                    chartInstance.State.is_chart = IN_cl_sl_s3_C;

```

```

                    chartInstance.Counters.il=0;
                }
            }

```

Figure 19A

```

break;
case IN_cl_s3_C:
/* During: C */
if((_sfEvent_temporal_example_ == event_CLK) &&
(chartInstance.Counters.il >= 5)) {
/* Exit: C */
/* Entry: A */
chartInstance.State.is_chart = IN_cl_sl_A;
chartInstance.Counters.il=0;
}
break;
}
}
}

```

1970

```

void broadcast_chart_CLK(void)
{

```

```

{
int8_T previousEvent;
previousEvent = _sfEvent_temporal_example_;
_sfEvent_temporal_example_ = event_CLK;
chart();
_sfEvent_temporal_example_ = previousEvent;
}
}

```

1900

Figure 19B

BEST AVAILABLE COPY

2000

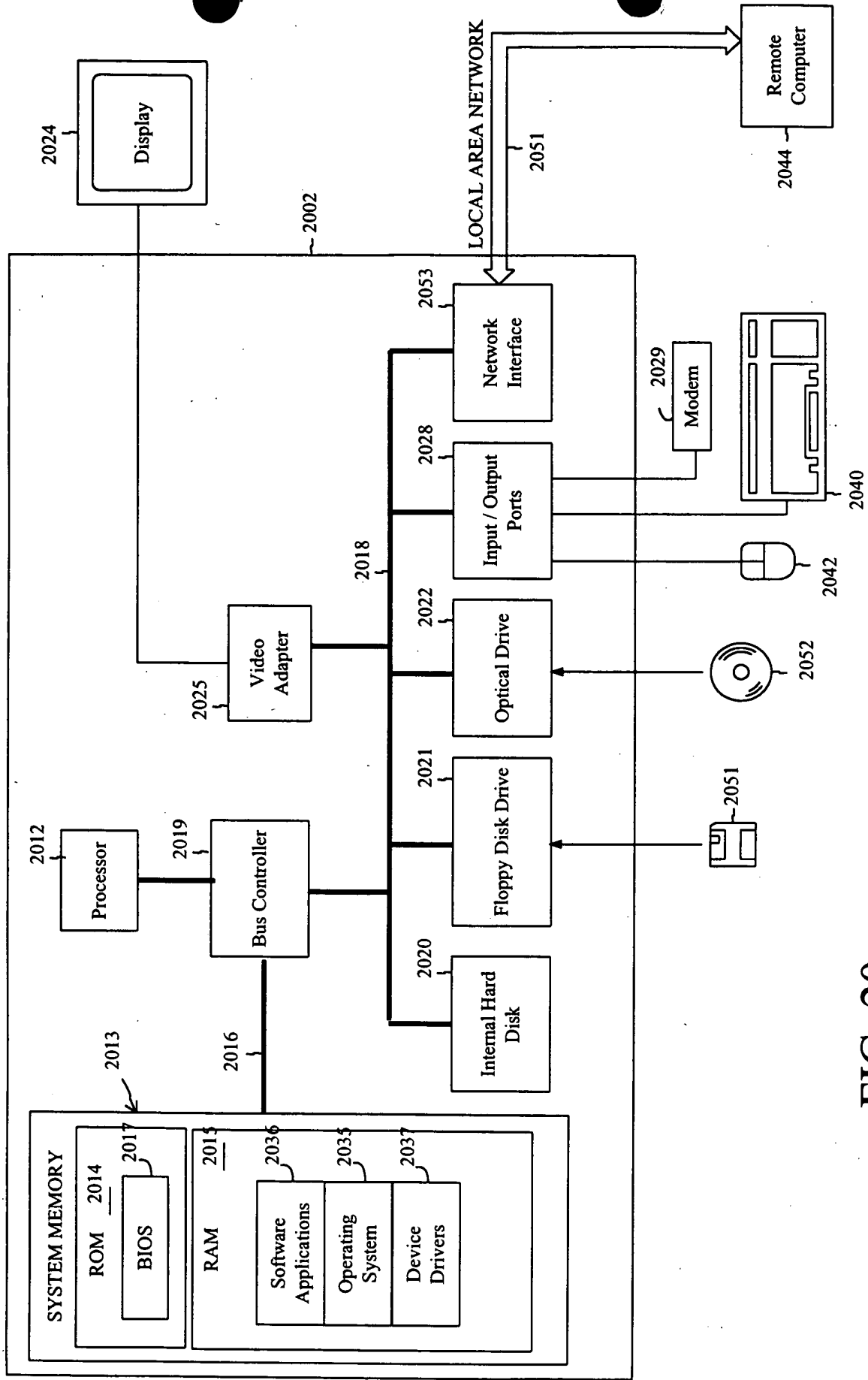


FIG. 20